The Factory Must Grow: Automation in Factorio

Kenneth N. Reid
Michigan State University
East Lansing, MI
Ken@kenreid.co.uk

Iliya Miralavy
Michigan State University
East Lansing, MI

Stephen Kelly
Michigan State University
East Lansing, MI

Wolfgang Banzhaf
Michigan State University
East Lansing, MI

Cedric Gondro
Michigan State University
East Lansing, MI

ABSTRACT

The video game Factorio has a myriad of gameplay scenarios which are analogous to real-world problems, and is a useful simulator for developing solutions for them. In this paper we define the logistic transport belt problem, we provide an interface to allow optimizers in any programming language to interact with Factorio, and we provide an initial benchmark. Results are presented for three different meta-heuristic techniques to optimize solutions to this novel problem.

KEYWORDS

Operational research, video games, meta-heuristics

ACM Reference Format:
Kenneth N. Reid, Iliya Miralavy, Stephen Kelly, Wolfgang Banzhaf, and Cedric Gondro. 2021. The Factory Must Grow: Automation in Factorio. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459463

1 INTRODUCTION

Factorio is a video game produced by Wube software[1], which was fully released in August 2020. Much like an ecology, Factorio can be thought of as a large set of varied interlinking problems and solutions, where balance is key. For example, focusing solely on production will likely increase the pollution output, cascading towards a game failure.

In this paper, we provide an overview of some such problems and an interface by which users can connect external optimizers to Factorio. As an example of this interface we tackle a small but common set of problems: logistic belt routing. Belts transport raw resources and manufactured goods between collection points, machinery and storage. With hundreds of items in the game which can be delivered this way, often requiring multiple belts dedicated to each item, this is both a complex and common problem facing players.

In a poster by Duhan et al [1] the concept of creating a reinforcement learning based engine to play Factorio is described, with the potential for creation of transferable knowledge to real-world scenarios, though no further work has yet been published.

In Factorio logistic belts are used to solve problems of transporting materials. A clear similarity between this contribution and real-world problems is the use of conveyor belts, their purpose, functionality and optimization. Conveyor belts, in the real world, have been referred to as productive and a particularly economical transport mechanism with high performance and a small carbon footprint compared to alternatives [2].

In this paper, we focus on the logistics belt placement problem. In order to represent various aspects of belt placement, we define 6 problem instances with varying degrees of difficulty. Each instance is represented as an integer matrix of size 3×3, 6×6 or 12×12 which defines the input and output location. The goal is to place transport belts such that the input and output are properly connected. Each size of matrix includes a version with and without obstacles. For a visual representation, see Fig 1.

Figure 1: 12 x 12 matrix with obstacles

The quality of a solution is measured through its fitness value. A fitness of 0 is assigned if a solution is deemed infeasible having failed one or more hard constraints, or a value between 0 and 1 otherwise. If a solution is feasible, then each soft constraint can be evaluated to return a single fitness value from the objective function. Fitness is calculated by considering the number of objects inserted into the system compared to the number output by the solution provided by the optimizer [4].

[1] https://factorio.com/

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459463

2 IMPLEMENTATION

We introduce Factorio Optimizer Interface (FOI)[2], an open-source cross-platform interface written in Python that uses Remote Console (RCON) protocol to enable problem solvers to connect and interact with a running Factorio server console. FOI incorporates file system and command-line interface, two programming language-independent methods, to communicate with problem solvers.

We utilize a version of Simulated Annealing (SA) with simultaneous solution comparisons known as Parallel Simulated Annealing (PSA). This is largely due to the inefficient nature of a serial SA within the constraints of this problem, as well as due to the proven ability of similar algorithms such as Differential Evolution to solve larger search spaces, for example by Whalen et al [5].

Genetic Programming (GP) is an evolutionary algorithm which evolves computer programs to solve a problem that can be computationally represented. We present a novel algorithm, q Genetic Programming (qGP), for the first time in this paper to solve the logistic belt problem. This algorithm represents the solutions with a linear genome consisting of a mixture of strings and integers that can be mapped to operators (functions or non-terminals) and operands (terminals).

Reinforcement Learning (RL) is a type of machine learning that simulates the way living organisms adapt through trial-and-error interaction with their environment over time. Many real-word decision-making problems can be modeled in this way, including game playing. In this work, we use a combination of GP and RL to develop generalized game-playing programs capable of solving multiple unique belt problems. No problem-specific functions are used. The particular GP framework employed for this study is Tangled Program Graphs (TPG) [3].

3 SUMMARY OF RESULTS

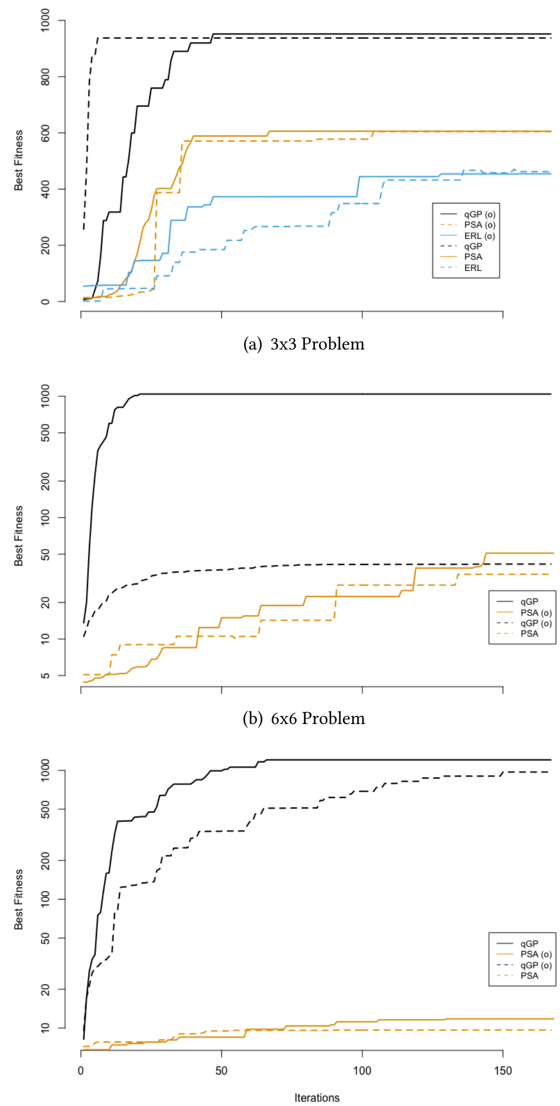
The video game Factorio provides an untapped suite of research problems, and this paper provides an initial step into exploring them. We present an introduction to problems which can be found within Factorio and define the logistic belt optimization problem. We provide exemplar algorithms to tackle the problem and compare their efficacy.

Our results show a variety of algorithm types and the benefits of exploratory algorithms such as Parallel Simulated Annealing, the strengths of domain-focused genetic programming and the flexibility of general purpose meta-heuristics through evolutionary reinforcement learning. Figure 2 plots the average best fitness of the best solution discovered by each algorithm over 167 iterations for each problem matrix. Due to the computational cost of evaluations in Evolutionary Reinforcement Learning (ERL), only the 3x3 matrix problems are considered. This work also provides an interface with Factorio to allow external optimization algorithms to connect to Factorio.

4 ACKNOWLEDGMENTS

This work was supported in part by the National Institute of Food and Agriculture (AFRI Project No. 2019-67015-29323), the BEACON Center for the Study of Evolution in Action and through computational resources and services provided by the Institute for Cyber-Enabled Research at Michigan State University.

[2]https://github.com/DrKenReid/FactorioBeltProblemGECCO



(a) 3x3 Problem

(b) 6x6 Problem

(c) 12x12 Problem

Figure 2: Average Best-Fitnesses by Iterations. (y-axis of 6x6 and 12x12 plots are log scale)

REFERENCES

[1] Duhan, S., Zhang, C., Jing, W., Li, M.: Factory optimization using deep reinforcement learning AI. Purdue Undergraduate Research Conference (2019), https://docs.lib.purdue.edu/purc/2019/Posters/57

[2] Grinčová, A., MArAsová, D.: Experimental research and mathematical modelling as an effective tool of assessing failure of conveyor belts. Eksploatacja i Niezawodność **16**(2) (2014)

[3] Kelly, S., Heywood, M.I.: Emergent solutions to high-dimensional multitask reinforcement learning. Evolutionary Computation **26**(3), 378–380 (2018)

[4] Reid, K.N., Miralavy, I., Kelly, S., Banzhaf, W., Gondro, C.: The factory must grow: Automation in factorio. arXiv preprint arXiv:2102.04871 (2021)

[5] Whalen, I., Banzhaf, W., Al Mamun, H.A., Gondro, C.: Evolving SNP Panels for Genomic Prediction. In: Banzhaf, W., Chen, B., Deb, K., Holekamp, K., Lenski, R., et al. (eds.) Evolution in Action: Past, Present and Future, pp. 467–487. Springer (2020)